

Problem statement:

Given an array w of positive integers, where $w[i]$ describes the weight of index i , write a function `pickIndex` which randomly picks an index in proportion to its weight.

Note:

1. $1 \leq w.length \leq 10000$
2. $1 \leq w[i] \leq 10^5$
3. `pickIndex` will be called at most 10000 times.

Example 1:

Input:

`["Solution","pickIndex"]`

`[[[1]],[]]`

Output: `[null,0]`

Example 2:

Input:

`["Solution","pickIndex","pickIndex","pickIndex","pickIndex","pickIndex"]`

`[[[1,3]],[],[],[],[],[]]`

Output: `[null,0,1,1,1,0]`

Explanation of Input Syntax:

The input is two lists: the subroutines called and their arguments. `Solution`'s constructor has one argument, the array w . `pickIndex` has no arguments. Arguments are always wrapped with a list, even if there aren't any.

My insights for random pick with weight questions:

We are given an array of weight let's assume $w=[1,2,3]$ and indexed as $w[0]$, $w[1]$, $w[2]$

We have to randomly pick an index and the probability of an index that should be chosen will depend on the percentage weight of the overall weight.

Let us say:

Probability of $w[0] = 1/(1+2+3)$

Similarly probability of $w[1] = 2/(1+2+3)$

And same for $w[3] = 3/(1+2+3)$

The percentage contribution of these indexes will $\frac{1}{6}$, $\frac{2}{6}$ and $\frac{3}{6}$ for index $w[0]$, $w[1]$ and $w[2]$ respectively.

Algorithm:

We are given with the weights and index

Index = [0] [1] [2]

Weight = [1, 2, 3]

Calculate the cumulative sum

Cumulative sum = 1 3 6

Create a range to check where exactly the randomly picked weight falls....each range is described as a bucket

Now lets create bucket for these [0-1], (1,3], (3, 6]

Here we will also need the total sum to know upto what range we should create a random number.

And the total_sum will be 6

Create a random number

Check the random number in the ranges of the bucket to do that we can either select the linear search method or binary search

Check the middle element

If the random number is less than the middle number go left , if not search on the right side

Output the index of the bucket range in which the number lies.